

A Database-Driven Ontology for German Grammar*

Roman Schneider

Abstract

The main objective of this article is to describe the current activities at the Mannheim Institute for German Language regarding the implementation of a domain-specific ontology for German grammar. We differentiate ontology bases from ontology management systems, point out the benefits of database-driven solutions, and go step by step through all phases of the ontology lifecycle. In order to demonstrate the practical use of our approach, we outline the interface between our ontology and the *grammis* web information system, and compare the ontology-based retrieval mechanism with traditional full text search.

1 Motivation

More than a decade ago, ontologies became a popular research topic in the fields of artificial intelligence, knowledge engineering, and information retrieval. Of course, the idea of describing relationships between real world objects and/or abstract topics is not that new, but often employed in different contexts and applications. Still today people often use the term “ontology” to mean different things, e. g., word nets, thesauri, or taxonomies.¹ Generally, there is little doubt about the importance and usefulness of domain-specific ontologies in contemporary knowledge representation environments. Modern information retrieval heavily relies on semantic add-ons for the classification and processing of distributed resources, and the popular vision of a future “semantic web”² will even force this trend. In order to establish language-independent frameworks, ambitious research activities within the knowledge engineering community deal with the modelling, coding, and linking of universal knowledge structures. Prominent examples of interdisciplinary developments are the Suggested Upper Merged Ontology (SUMO), Cyc/OpenCyc, the Generalized Upper Model (GUM) or DOLCE/WonderWeb.³

On top of these upper ontologies as well as stand-alone, more and more domain-specific ontologies are under construction. They codify concepts and relationships for single areas of

* Published in: *Data Structures for Linguistic Resources and Applications*, Georg Rehm, Andreas Witt, Lothar Lemnitzer (eds.), Tübingen: Gunter Narr Verlag. 2007. pp. 305–314.

¹ See, e. g., Schneider (2006) for a more detailed definition.

² See Berners-Lee et al. (2001), but also up-to-date online resources like <http://www.mindingtheplanet.net>.

³ Publication lists and technical information about these projects can be found at <http://www.ontologyportal.org>, <http://www.opencyc.org>, <http://www.purl.org/net/gum2> and <http://wonderweb.semanticweb.org>.

interest, allow visualization and browsing of structures, and often include the goal of automated reasoning. For example, categories and relations dedicated to descriptive linguistics are captured with the help of GOLD (General Ontology for Linguistic Description), which is built on top of SUMO.⁴ However, even when limited to certain domains, ontology authors are faced with the simple fact that the terminological use of concepts varies between terminological systems. This seems especially true for linguistics, where different theories, schools, or even authors often not only name concepts differently, but even assign varying meanings to identical terms. For example, generative grammars usually regard a verb phrase or even a sentence as a phrase (complementizer phrase), whereas others – like the *Grammar of German Language* (Zifonun et al., 1997) published at the Institute for German Language in Mannheim – do not. Varying theories, varying timelines, varying analyzing criteria – creating a backbone hierarchy here is definitely no pleasing job.

The heterogeneous use of terminology not only confuses human readers, but in the case of digitization makes information exchange between software systems as well as human-computer interaction more difficult. Taken into consideration that ontologies are often seen as enabling technology for information sharing, they should cope with these difficulties. A semantically enriched retrieval application for the exploration of large linguistic corpora should “know” about theory-related details so that it can offer appropriate solutions. Beisswenger et al. (2004) introduce a way to deal with terminological differences and similarities. In order to bring together different systems, they model a terminological wordnet (TermNet) which subsumes similar concepts under so-called “termsets” and thereby expands the synset paradigm used by the Princeton WordNet or its German equivalent GermaNet.⁵ We will point out later how we incorporate this idea into our ontology model.

Our primary motivation for building the *Ontology of German grammar* was to improve information retrieval, content exploration, and text classification for the *grammis* web information system.⁶ Work on *grammis* is under way since the mid-nineties at the Institute for German Language (Institut für Deutsche Sprache, IDS) in Mannheim, Germany. Today it provides comprehensive information about German grammar, using hypertext and multimedia techniques. Currently five core components, administered within an object-relational database management system, can be used online: 1. the “systematic grammar” (Systematische Grammatik) tries to sketch an overall view of German grammar. 2. A grammatical FAQ (Grammatik in Fragen und Antworten) answers selected questions that exemplify more general problems. 3. A terminological dictionary (Terminologisches Wörterbuch) describes technical terms; 4. the grammatical dictionary (Grammatisches Wörterbuch) comprises function words, affixes and – shortly – selected verbs. 5. The system is rounded off by a grammatical bibliography (Bibliografie zur deutschen Grammatik, BDG), which up to now uses a flat keyword index.⁷

⁴ See, e.g., Farrar et al. (2002) and <http://www.linguistics-ontology.org>.

⁵ See Fellbaum (1998) as well as Kunze and Lemnitzer (2002), respectively.

⁶ Schneider (2004); online at <http://www.ids-mannheim.de/grammis/>.

⁷ Latest print version is Frosch et al. (2003); next publication is scheduled for 2007.

2 Ontology Management Systems and the Ontology Lifecycle

Even within the knowledge engineering community, the term “ontology” is used in different contexts. Discussions about formalizing ontology models or about cognitive adequateness of knowledge structures primary refer to internal representation models, i. e., to the logic of organizing and linking of single knowledge fragments. Questions concerning the correlation between symbol, concept, and referenced entity – thus: concerning the meaning of stored concepts – play a minor role in this discussion. But the latter take center stage when talking about adequateness of terminological description, i. e., about the reasonable covering of domains using a certain vocabulary. Finally, IT-related questions become crucial, because administration of large ontologies needs a sophisticated storage infrastructure.

Against this background, and for the sake of accurate definition, we differentiate ontology bases from ontology management systems. An ontology consists of an ontology base (sometimes also referred to as knowledge base), which is a consistent formal description of concepts (also: classes). Attributes (also: slots) are inherited from general to more specific concepts. Mutual relationships of any kind can be modelled by explicitly named relationship types.

Efficient administration of this ontology base as well as the management of query interfaces and analyzing modules should be carried out by an ontology management system. It assures logical consistency of the coded statements, and provides functionalities for navigation and retrieval. Furthermore, it integrates inference mechanisms and allows interoperability, i. e., content mapping and merging. The broad R&D linecard goes from file-based solutions⁸ to distributed systems that rely on mature databases. The former offer out of the box access to ontological data, while database-driven platforms score with scalability, robustness, data security, and performance. Ambitious future application scenarios for the semantic web will probably require large-scale ontologies with a complexity close to those of AI expert systems.

The aspect of complexity attracts attention not only while choosing the appropriate management system. Furthermore, it limits the suitability of ad hoc solutions during the production process. In order to coordinate different working stages and to increase product quality, software engineering guidelines propose the use of specific process models. This is due to the fact that software engineering should not be seen simply as “creative process”, but needs a structured framework for proper planning and to be less error-prone in the long run. With the help of such models, ontology development can be optimized just as well. We suggest the use of a lifecycle model, which considers the complete ontology authoring activity as a continuing process and breaks it up into component subprocesses. These steps, defined by specific entry and exit criteria, can be passed through several times for the purpose of gradual refinement.

A generic ontology lifecycle can be divided roughly into the following phases: 1. production 2. monitoring 3. revision 4. release 5. distribution/integration in applications. Thus, the starting point of ontology development is the production of concepts and relationships, which

⁸ For example, ontology editors and environments such as Protégé (<http://protege.stanford.edu>), Ontolingua (<http://www.ksl.stanford.edu/software/ontolingua/>), and WebOnto (<http://kmi.open.ac.uk/projects/webonto/>), that store content directly in OWL or DAML+OIL format in the file system. Of course, these easy-to-use authoring tools can also be integrated in more complex solutions.

again can be divided into subprocesses: situation analysis, objective formulation, determining relationship types and concepts etc. Subsequently, an examination regarding logical consistency and terminological adequateness should take place. Optionally, the contents are revised before the public release. All phases following the initial production can be seen as iterative processes, since probably every ontology needs to be examined, corrected and/or refined several times.

3 Modelling Relationships

Concepts can be connected – permanently, temporarily or situationally – by most different semantic relations. Relationship types systematize these relations by using logical characteristics like reflexivity, symmetry, or transitivity.⁹ In order to bring together theoretical desiderata with practical demands and limitations, we combine well-established principles of ontological engineering – e. g., the use of standard hyponymy/meronymy relationship types like Broader Term Generic (BTG) or Broader Term Partitive (BTP)¹⁰ – and modelling concepts already tested in state-of-the-art applications.

As already mentioned, Beisswenger et al. (2004) introduce termsets for the connection of similar terminological concepts. We stick to this idea, but expand the model by adding some theory-related attributes and, secondly, allowing the explicit linking of individual concepts belonging to different termsets. Figure 1 illustrates our model. It contains three termsets, indicated by dotted border lines. The bottom termset contains the two concepts {Verbgruppe} and {Verbalphrase}, recognizable by rectangles with rounded corners. {Verbgruppe} is characterized by a theory-related attribute named “IDS”, meaning that it is used primarily when referring to the IDS Grammar of German Language. The concept {Verbalphrase} consists of four lexical entries: 1. {Verbalphrase} with a PT-marker for Preferred Term and with a language attribute (German). 2. {Verbphrase} linked to the former by a synonymy (SYN) relation. 3. {VP} linked by a abbreviation (AB) relation. 4. {Verb Phrase} with a language attribute (English) and linked with a translation (TR) relation. The complete termset, which additionally may be characterized by an optional and inheritable attribute for the grouping of co-hyponyms, is linked with its hyperonym termset by a BTG relation.

In order to clarify the benefit of linking not only termsets, but also individual concepts, our example illustrates the relationships between {Phrase} (engl. “phrase”) and {Satz} (engl. “sentence”). Basically, the corresponding termsets are connected with the help of a Broader Term Partitive (BTP) relation (meronymy). Beyond this, since generative grammars usually classify sentences (complementizer phrases) as phrases, only these two concepts – singled out by a theory-related attribute – are linked by an Narrower Term Generic (NTG) relation (hyponymy). This fact, explicitly coded within the ontology base, should facilitate communication between people or computer systems using different terminological vocabularies.

Furthermore, we use standard relationship types like Related Term (RT) for the linking of termsets that are associated in some way, but without the necessity of deeper relationship explanation. Good examples are {Wortschatz} (engl. “vocabulary”) and {Wortschatzerweiterung} (engl. “vocabulary extension”) or {Fokus} (engl. “focus”) and {Fokuspartikel} (engl. “focusing

⁹ See, e. g., the detailed analysis by Lehmann (1996).

¹⁰ Compliant with both the ISO-2788 and ANSI Z39.19 standards.

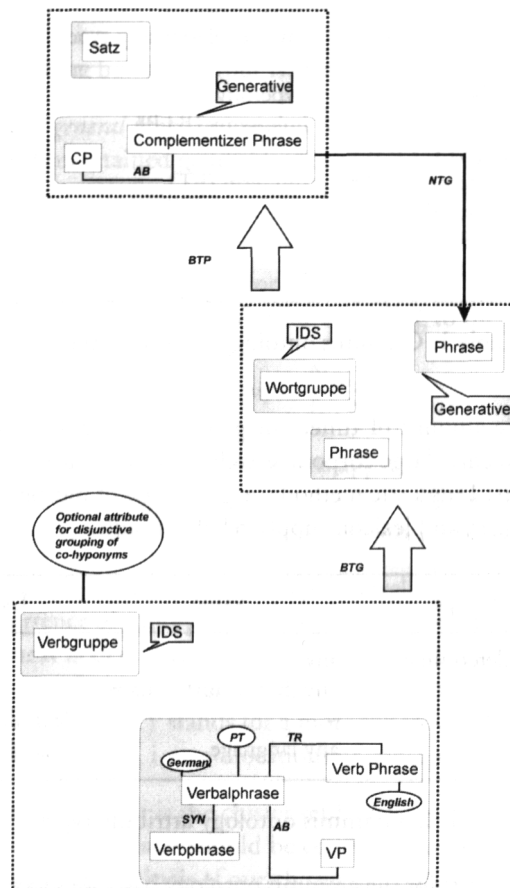


Figure 1: Grammis ontology modelling structure

adjunct”): Focusing adjuncts like “sogar”, “selbst”, “nur” mark the focus. Because we do not see a need for introducing a special relationship type for this, we simply call them RTs. Table 1 reflects the relationship types, table 2 displays the list of attribute types.

4 Detecting Concepts

Apart from modelling relationship types, the selection of the concepts – i. e., filling up the ontology base with content – is probably one of the most challenging subtasks within the ontology lifecycle. Concepts are chosen because of their relevancy to express the knowledge in a given domain, and can be discovered by three different approaches: 1. Intellectual/manual compilation of all relevant domain concepts by human experts. 2. Use of statistical methods on a given representative corpus. 3. Use of linguistic methods. Usually, the selection depends primarily on project-specific factors, preferences, and objectives. Recourse to human knowledge

Relationship type	Inverse relationship type	
hyponymy (NTG)	hyperonymy (BTG)	conceptual
meronymy (NTP)	holonymy (BTP)	conceptual
related term (RT)	–	conceptual
abbreviation/acronym (AB)	full term (FT)	lexical
spelling variant (SP)	–	lexical
translation (TR)	–	lexical
synonymy (SYN)	–	lexical

Table 1: Grammis ontology relationship types

demands a relatively large amount of time, but generally guarantees high quality. Statistical methods depend on sufficiently large corpora as well as on long-time experience in fine tuning algorithms and parameters. Linguistic methods, e. g., the use of morpho-syntactic information, succeed only if parser, tagger, and lexicon supply reliable results.

Attribute type	Value	Used for
Co-hyponym determination	any	termset
Theory	any theory/author name	concept
Preferred term (PT)	–	lexicalization/term
Language	any language	lexicalization/term

Table 2: Grammis ontology attribute types

For the detection of concepts for the grammis ontology base, we successfully used a combined method comprising statistical exploration, linguistic analysis as well as manual post-editing. The underlying specialist language corpus was made up of XML-structured hypertexts from the grammis and ProGr@mm information systems hosted at IDS. Altogether we included a total of about 2,000 hypertext nodes with almost 1,000,000 wordforms (N_{SL}). Furthermore, we used COSMAS (Corpus Search, Management and Analysis System, <http://www.ids-mannheim.de/cosmas2/>) for exploring 160 general language corpora with more than 1.6 billion wordforms (N_{GL}). In the following, we present our six steps for concept acquisition:

1. *Frequency analysis of specialist language corpus:* The specialist language (SL) hypertexts are used as input. We tokenize the corpus and collect frequency information for each token (f_{SL}). Stop words like “und”, “aber” etc. are omitted. Wordforms with a frequency value below a previously defined threshold are filtered out. Output is an ordered list with two columns (wordform, f_{SL}).
2. *Markup analysis:* We use the output list from step 1 as well as XML-coded meta information¹¹ from the grammar corpus as input. Wordforms appearing in the most prominent

¹¹ A description of grammisML, which is used here, can be found in Schneider (2004, p. 251 ff.).

hypertext structures – i. e., in titles, subtitles, definitions, and semantically typed hyperlinks – receive a ranking bonus. Output is an accordingly modified f_{SL} list.

3. *Frequency analysis of general language corpus:* We use the output list from step 2 together with the COSMAS-maintained general language (GL) corpora as input. For each wordform, we calculate the GL-frequency value (f_{GL}). Output is a list with three columns (wordform, modified f_{SL} , f_{GL}).
4. *Weirdness value:* We use the output list from step 3 as input. With the help of a well-tested algorithm¹², we compute a “weirdness” value according to the following equation:

$$\tau(w) = \frac{N_{GL} f_{SL}}{f_{GL} N_{SL}} \quad (1)$$

The computed value tells us which wordforms appear significantly more frequent in the specialist corpus than in the general language corpus. Higher values indicate interesting wordforms, i. e., concept candidates. Wordforms with a low value are filtered out.

5. *Collocation analysis:* We use the list from step 4 as well as the SL-corpus as input. We examine the co-occurrence of concept candidates by using varying environments (sentences, paragraphs, hypertext nodes). Even basic vectors can be detected: given that concept candidate X appears more frequent in conjunction with concept candidate Y than Y together with X, then we may say that Y stands for a more general concept than X. Output is a set of concept candidate clusters, i. e., collocations of concept candidates.
6. *Relationship assignment:* Input is the cluster set from step 5. Now a human expert has to decide which concept candidates should be considered as domain-specific and which relations could be coded on the basis of our cluster set. Output is a tentative terminological net, which already contains some partial hierarchies.

5 Database Implementation and Retrieval

When it comes to database implementation, the number of possible modelling strategies, methods, and systems is enormous. Assuming that reliable and high-performance ontology management solutions preferably require professional database management systems (DBMS)¹³, we decided to adopt the object-relational DBMS already in use for grammis. For portability reasons, we designed our conceptual data model according to the well-established entity-relationship (ER) paradigm, and used the relational approach for database implementation. Figure 2 shows our model, based on the example ontology structure from section 3. The further implementation process is quite straightforward.

Obviously, a major benefit of using integrated ontologies is their support for text classification and retrieval. Traditional full text search, based on the vector model, is limited in terms of semantic markers. Most users find it difficult to formulate queries which are well designed

¹² See the comprehensive description in Gillam et al. (2005).

¹³ For a discussion about storage options see, e. g., Schneider (2004, p. 204 ff.).

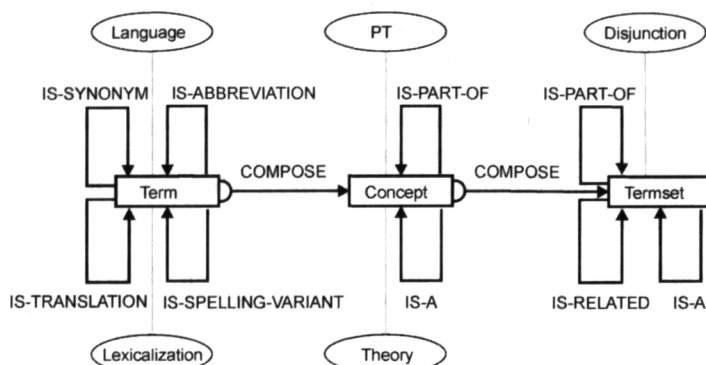


Figure 2: Conceptual database model

for retrieval purposes. Nevertheless, users of complex information systems often consider full text search as the preferred access option. But it supplies satisfying results only if humans and computer speak the same language, i. e., share a common terminology. For grammis this means: if the user types in “Ergänzung”, the system should realize that this is synonym for “Komplement” (engl. “complement”), and it should link it to “Valenz” (engl. “valency”). The query is expanded, and the result set increases. In order to avoid a disproportional increase, on a certain level the reverse strategy of query reformulation seems necessary: if the system recognizes that a search term ranks high in the ontological hierarchy, e. g., “Valenz”, it should offer a set of subordinated terms, e. g., “Verbvalenz”, with probably less retrieved documents.

A graphical representation of the ontology structure assists the ontology author through all phases of the ontology lifecycle. Besides, it helps end users in situations when they cannot precisely formulate their information need or just want to browse the whole system. For these reasons, grammis includes a graphical retrieval and navigation frontend. Figure 3 illustrates the functionality: in the center we see the currently accessed termset. Above, bordered by specifically colored block elements and serving as hyperlink anchors, the immediately superordinated hyperonymes and holonymes can be found; below are hyponymes and meronymes. Associated concepts (RTs) are displayed also. By pointing and clicking, users can activate the different relations and change their position within the informational space.

Since our database-driven ontology is directly connected to the whole grammis information system, the frontend comprises appropriate retrieval options, mapping user input to standard SQL (Structured Query Language) statements. By drag-and-drop, users are allowed to insert any term from the graphical structure into one of the three containers on the right side. The system then sifts through the hypertext base as well as through the bibliography and all dictionaries. The number of hits is immediately displayed next to the container; the actual result set is presented by request in a separate pop-up window. Results of combined queries are shown between the containers.

Bibliography

- Beisswenger, Michael; Storrer, Angelika and Runte, Maren (2004): "Modellierung eines Terminologienetzes für das automatische Linking auf der Grundlage von WordNet". *LDV-Forum* 19 (1/2): pp. 113–125.
- Berners-Lee, Tim; Hendler, James and Lassila, Ora (2001): "The Semantic Web". *Scientific American* <http://www.scientificamerican.com/2001/0501issue/0501bern timers-lee.html>.
- Butt, Miriam (editor) (2006): *Proceedings of KONVENS 2006 (Konferenz zur Verarbeitung natürlicher Sprache)*. Konstanz.
- Farrar, Scott; Lewis, William D. and Langendoen, D. Terence (2002): "A Common Ontology for Linguistic Concepts". *Proceedings of the Knowledge Technologies Conference* <http://www.emeld.org/documents/KnowTech-CommonOntology.pdf>.
- Fellbaum, Christiane (1998): *WordNet: An Electronic Lexical Database*. Cambridge, MA: The MIT Press.
- Frosch, Helmut; Schneider, Roman; Strecker, Bruno and Eisenberg, Peter (2003): *Bibliographie zur deutschen Grammatik. 1994–2002*. Tübingen: Gunter Narr Verlag.
- Gillam, Lee; Tariq, Mariam and Ahmad, Khurshid (2005): "Terminology and the Construction of Ontology". *Terminology* 11 (1): pp. 55–81.
- Kunze, Claudia and Lemnitzer, Lothar (2002): "Germanet – Representation, Visualization, Application". *Proceedings of LREC 2002*.
- Kunze, Claudia; Lemnitzer, Lothar; Lünen, Harald and Storrer, Angelika (2006): "Modellierung und Integration von Wortnetzen und Domänenontologien in OWL am Beispiel von GermaNet und TermNet". In: Butt (2006), pp. 91–96.
- Lehmann, Christian (1996): "Linguistische Terminologie als relationales Netz". In: *Nomination – fachsprachlich und gemeinsprachlich*, edited by Knobloch, Clemens and Schaefer, Burkhard, Opladen: Westdeutscher Verlag, pp. 215–267.
- Schneider, Roman (2004): *Benutzeradaptive Systeme im Internet: Informieren und Lernen mit GRAMMIS und ProGr@mm*. Mannheim: IDS. Amades, Band 4/04.
- Schneider, Roman (2006): "Eine Ontologie für die Grammatik. Modellierung und Einsatzgebiete domänenspezifischer Wissensstrukturen". In: Butt (2006), pp. 125–129.
- Seewald-Heeg, Uta (2006): "Terminology Exchange without Loss? Feasibilities and Limitations of Terminology Management Systems (TMS)". *LDV-Forum* 21 (1): pp. 5–18.
- Zifonun, Gisela; Hoffmann, Ludger; Strecker, Bruno; Ballweg, Joachim; Brauße, Ursula; Breindl, Eva; Engel, Ulrich; Frosch, Helmut; Hoberg, Ursula and Vorderwülbecke, Klaus (1997): *Grammatik der deutschen Sprache*. Berlin: de Gruyter. Schriften des Instituts für Deutsche Sprache, Band 7.